

00/60/90
1c555 U.S. PTO
06/09/00

EXPRESS MAIL NUMBER: EL389646768US

DATE OF DEPOSIT: June 9, 2000

I hereby certify that this paper is being deposited with the United States Postal Service "EXPRESS MAIL Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to: Box PATENT APPLICATION, Assistant Commissioner for Patents; Washington, DC 20231.

Patricia K. Parry 6/9/00
Patricia K. Parry

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

* * *

UTILITY APPLICATION TRANSMITTAL LETTER

ASSISTANT COMMISSIONER FOR PATENTS
Washington, D.C. 20231

ATTN: BOX PATENT APPLICATION

Sir:

Transmitted herewith for filing is the patent application of:

INVENTOR(s): Chris Rygaard

FOR: MOBILE APPLICATION SECURITY SYSTEM AND METHOD

Enclosed are:

- [28] pages of specification
- [6] pages of claims
- [1] page of abstract
- [19] sheets of informal drawings
- [] Declaration and Power of Attorney (executed)
- [] Assignment of the application with Recordation Cover Sheet
- [] Information Disclosure Statement
- [] Preliminary Amendment
- [] Small Entity Declaration
- [X] Other: acknowledgement postcard.

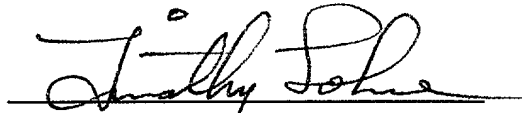
All future correspondence should be addressed to:

Timothy W. Lohse
GRAY CARY WARE & FREIDENRICH LLP
Attn: Patent Department - HV
400 Hamilton Avenue
Palo Alto, CA 94301
(650) 320-7426

Respectfully submitted,

GRAY CARY WARE & FREIDENRICH LLP

Dated: June 9, 2000



Timothy W. Lohse
Reg. No. 35,255
Attorney for Applicant

GRAY CARY WARE & FREIDENRICH LLP
Attn: Patent Department - HV
3340 Hillview Avenue
Palo Alto, CA 94304

MOBILE APPLICATION SECURITY SYSTEM AND METHOD

Background of the Invention

This invention relates generally to a system and method for enhancing the operation and security of a software application and in particular to a system and method for improving the security of a mobile software application.

In traditional computing systems, communication between computers is either code (a software application) or data (a file containing information) and there is no notion of a program moving between hosts while it is being executed. Thus, with a typical computing system, a person may execute a software application (e.g., Microsoft Word) on his own computer and then forward the results of the execution of the software application (e.g., a Word document) to another user. The other user may then view the Word document by executing his own copy of Microsoft Word. A user may also send another user an executable software application file that the other user may download and execute on his own computer. However, these traditional computing systems do not recognize a single instantiation of a software program that may be executed by one or more different computers in order to complete the execution of the software application.

A mobile application, sometimes also called a mobile app or a mobile agent, is a currently executing computer software application/program, or part of a currently executing computer program that can physically move from one computer to another (between hosts) while it is being executed: A mobile application's software may or may not have been previously

installed on a particular computers prior to the arrival of the mobile application. The mobile applications are said to jump from one computer to another computer and the process of jumping from one computer to another computer is also referred to as a jump.

The process of initiating a jump between computers is commonly known as a dispatch.

5 Typically, each mobile application will carry with it an ordered list or tree of hosts which the mobile application must visit during its execution, and such a list or tree is called the mobile application's itinerary. An example of a mobile application and its itinerary is described below with reference to Figure 2. The computers that can receive and dispatch mobile applications are called hosts. The collection of hosts, computer networks, and software which executes and
10 supports the mobile applications, and the mobile applications themselves, is called the mobile application system.

A mobile application typically has at least two parts: the state and the code. The state of the mobile application contains all of the data stored, carried, and/or computed by the particular mobile application. The code of the mobile application is the set of computer instructions which
15 the host computer is intended to carry out on behalf of the mobile application during the execution of the mobile application by the particular host computer. In addition, a mobile application may have other parts, including an Access Control List (ACL), an itinerary, a datastore, an audit log, etc.

The problem faced by software products that support mobile applications are
20 insurmountable security problems. In particular, there are three problems that are most often cited:

1) An hostile host can send code with undesirable behavior to another host. Currently, there is no way to ensure that an hostile host cannot inject unsafe code into the mobile application system.

2) A mobile application cannot be protected from a hostile host. In particular, when a mobile application arrives at a host and begins execution, that mobile application is at the mercy of the host. In other words, there is no guarantee that the host will execute the computer instructions properly. There is not even any guarantee that the host will run any particular software at all; and

3) A mobile application cannot be securely sent to or received from a host outside of a group of trusted computers, known as the Trusted Computing Base (TCB).

A Trusted Computing Base (TCB) is the collection of computers, computer peripherals, and communication networks which must perform all requested operations properly, and must not perform extraneous operations, and are trusted to do so, in order to properly complete whatever computations are required.. A host outside of the TCB can perform nefarious tasks on the mobile application. This nefarious behavior cannot be controlled, and it cannot be detected. Therefore, once a mobile application has visited an untrusted host, it could be altered in an undesirable way, and therefore is a security hazard. In addition, the mobile application that visited the untrusted host can no longer be trusted to execute within the TCB. All of these security problems with mobile application need to be overcome before mobile applications become more accepted as a alternative to traditional computing systems. Thus, it is desirable to provide a mobile application security system and method that overcomes the above problems and

limitations with conventional mobile application systems and it is to this end that the present invention is directed so that mobile applications may be used in most financial, commercial, and military computer systems.

Summary of the Invention

5 The mobile application security system and method increases the overall level of security in using a mobile application. In a preferred embodiment, the system may use a client/server architecture wherein each host of a mobile application is treated as a client and a central computer is treated as the server. In operation, any time that a mobile application is going to jump between hosts, it must first pass through the central computer so that the central computer 10 may perform various security checks. The security checks ensure that the security of the mobile application is not compromised and overcomes the above problems with typical mobile application systems. In accordance with the preferred embodiment of the invention, the security system in accordance with the invention may detect unwanted changes in the code of the mobile application by comparing the mobile application received from the sending host with a copy of 15 the mobile application in the central computer. This ensures that a host cannot accidentally or purposely inject some unwanted code, such as a virus, into the mobile application. In accordance with another embodiment of the invention, the security system may prevent hostile or untrusted hosts from transmitting code to the other hosts in the mobile application system. In accordance with yet another embodiment of the invention, the security system may prevent unwanted 20 changes to the code of the mobile application. In yet another embodiment, the system may

prevent unwanted changes in the itinerary of the mobile application. In yet another embodiment, the system may prevent untrusted hosts from initially launching mobile applications.

Thus, in accordance with the invention, a mobile application security system and method are provided wherein the system comprises a central computer for controlling the security of a mobile application system; one or more host computers connected to the server computer wherein each host computer executes the mobile application that jumps between the hosts during execution. The central computer further comprises means for monitoring the security of the mobile application as it jumps between the host computers wherein when the mobile application is communicated from a first host to a second host, it passes through the central computer. In accordance with one embodiment of the invention, the security monitoring further comprises means for detecting unwanted changes in the code associated with the mobile application when the mobile application is jumping between hosts.

In accordance with another embodiment of the invention, the security monitoring further comprises means for preventing a host from transmitting hostile code in a mobile application to another host. In accordance with yet another embodiment of the invention, the security monitoring further comprises means for detecting unwanted changes in the state of the mobile application. In accordance with yet another embodiment of the invention, the security monitoring further comprises means for detecting unwanted changes in the itinerary of the mobile application. In accordance with yet another embodiment of the invention, the security monitoring comprises means for preventing untrusted hosts from initially launching mobile applications.

Brief Description of the Drawings

Figure 1 is a diagram illustrating a typical mobile application and its operation;

Figure 2 is a diagram illustrating an example of a typical mobile application;

Figure 3 is a diagram illustrating the movement of a mobile application in a conventional
5 peer-to-peer mobile application system;

Figure 4 is a diagram illustrating a client/server mobile application security system in
accordance with the invention;

Figure 5 is a diagram illustrating the operation of the mobile application security system
of Figure 4;

10 Figure 6 is a diagram illustrating more details of the mobile application security system
shown in Figure 5;

Figure 7 is a diagram illustrating an example of the process for never retrieving code
from an untrusted host;

15 Figure 7a is a diagram illustrating a first embodiment of the mobile application security
system for detecting unwanted changes to the code of a mobile application in accordance with
the invention;

Figure 8 is a diagram illustrating a first example of a second embodiment of the mobile application security system for preventing hostile hosts from transmitting code to other hosts in accordance with the invention;

Figure 9 is a diagram illustrating a second example of a second embodiment of the mobile application security system for preventing hostile hosts from transmitting code to other hosts in accordance with the invention;

Figure 10 is a diagram illustrating a third example of a second embodiment of the mobile application security system for preventing hostile hosts from transmitting code to other hosts in accordance with the invention;

Figure 11 is a diagram illustrating a fourth example of a second embodiment of the mobile application security system for preventing hostile hosts from transmitting code to other hosts in accordance with the invention;

Figure 12 is a diagram illustrating a third embodiment of the mobile application security system for detecting unwanted changes to the state of a mobile application in accordance with the invention;

Figure 13 is a diagram illustrating a first example of a fourth embodiment of the mobile application security system for detecting unwanted changes in the itinerary of the mobile application in accordance with the invention;

Figure 14 is a diagram illustrating a second example of a fourth embodiment of the mobile application security system for detecting unwanted changes in the itinerary of the mobile application in accordance with the invention;

Figure 15 is a diagram illustrating a third example of a fourth embodiment of the mobile application security system for detecting unwanted changes in the itinerary of the mobile application in accordance with the invention;

Figure 16 is a diagram illustrating a first example of a fifth embodiment of the mobile application security system for preventing untrusted hosts from launching a mobile application in accordance with the invention;

Figure 17 is a diagram illustrating a second example of a fifth embodiment of the mobile application security system for preventing untrusted hosts from launching a mobile application in accordance with the invention; and

Figure 18 is a diagram illustrating a third example of a fifth embodiment of the mobile application security system for preventing untrusted hosts from launching a mobile application in accordance with the invention.

Detailed Description of a Preferred Embodiment

The invention is particularly applicable to a client-server based mobile application security system and it is in this context that the invention will be described. It will be

appreciated, however, that the system and method in accordance with the invention has greater utility since it may be used with web-based systems for example.

Figure 1 is a diagram illustrating a typical mobile application 18 and its operation. In particular, the mobile application may start its execution on a first computer 20. At some point, the mobile application 18 is instructed to move to a second computer 22 and the mobile application jumps to the second computer. Once at the second computer, the mobile application resumes its execution on the second computer. At some later time, the mobile application is instructed to move to a third computer 24 and the mobile application jumps to the third computer and resumes its execution on the third computer. In this manner, the mobile application can execute on one or more different computers at different times. To understand the concept of a mobile application, an example of a typical mobile application will now be provided.

Figure 2 is a diagram illustrating an example of a typical mobile application and in particular, an intelligent expense report form. In this example, the mobile application facilitates the expense report process by automatically performing some functions. In particular, a salesman at a laptop computer 26 may initially fill out an expense report form and click OK when the expense report is ready. Automatically, the mobile application then sends itself to a manager's computer 28 for approval by the manager. In this example, the manager finds a problem with the form and returns it to the salesman so that the form automatically sends itself back to the salesman for an update. Next, the salesman makes the necessary corrections and clicks OK to send it automatically back to the manager. With the further updates, the manager accepts the expense form and clicks "OK". The mobile expense report form then automatically

sends itself to a computer 30 in the administration department. The mobile expense form then executes on the administration computer and updates a database 32 with the new information in the expense form. Next, the mobile expense report automatically sends itself to a computer 34 of the accountant. The mobile expense report then automatically starts to execute on the

5 accountant's computer and notifies the accountant that a check is needed so that the accountant can cut the check for the salesman. Thus, the mobile application has automated much of the expense report submission process so that the people involved in the process do not have to worry about ensuring that the expense report is approved. To better understand the problems associated with the typical mobile application, an example of the movement of the typical mobile

10 application will be described in more detail.

Figure 3 is a diagram illustrating the movement of a mobile application 40 in a conventional peer-to-peer mobile application system 42. In this example, the system 42 may include one or more host computers, such as Host1, Host2, Host3, Host4 and Host5, that execute the mobile application are different times as the mobile application jumps between the hosts as is well known. As shown in Figure 3, the mobile application 40 may jump directly from one host to another host such that there is never a central repository for information about the mobile application. Thus, a noted problem with the mobile application from Host 1 may never be known by the other Hosts. In addition, any of the Hosts in the system 42 may sabotage or alter the mobile application to perform some nefarious act, such as placing a virus into the mobile

15

20 application. It is desirable to provide a system wherein the hosts and the mobile application are protected from attacks and the invention solves these problems as will now be described.

Figure 4 is a diagram illustrating a client/server mobile application security system 50 in accordance with the invention. In particular, the system may include a server computer 52 and one or more host computers 54, such as Host 1, Host 2 and Host N, that may be connected to the server computer by a computer network 56, such as a wide area network, the Internet, the World Wide Web, a telephone line and a modem or the like. The computer network permits the server and hosts to communicate data between each other. Each host may be a typical computer system that includes a CPU and a memory for executing a software application such as a mobile application.

The server 52 may include a CPU 58 and a memory 60 along with a persistent storage device (not shown) for permanently storing one or more software applications or modules that may be executed by the CPU by loading the software applications or modules into the memory. The server may also include a database 62 that stores one or more mobile applications along with information about the mobile applications as described below. As shown, the memory of the server has a mobile application controller module 64 stored in it that, when executed by the CPU, control the security of the mobile applications and hosts as described below. In a preferred embodiment, the mobile application controller 64 may be one or more software application or modules, but the controller may also be implemented using hardware.

In a preferred embodiment, the mobile application controller 64 may include security software 66 and a communications software 68. The combination of the software may solve the problems with typical mobile application systems so that: 1) An hostile host cannot send code with undesirable behavior to another host; 2) A mobile application cant be protected from a

hostile host; and 3) A mobile application can be securely sent to or received from a host outside of a group of trusted computers, known as the Trusted Computing Base (TCB) without fear of hostile activity. The way in which the security system in accordance with the invention overcomes these problems will now be described.

5 Figure 5 is a diagram illustrating the operation of the mobile application security system 50 of Figure 4. In particular, the security system 50 in accordance with the invention uses a client/server based security model as opposed to the typical peer-to-peer arrangement as shown in Figure 3. Thus, using the security system 50 in accordance with the invention, there is
10 centralized server 52 which is not a host for the mobile applications, but acts as a server for the participating hosts (Host1, Host2, Host3, Host4 and Host 5 in this example) that are the clients. Thus, in accordance with the invention, each of these clients (Hosts) communicates with only the server and never directly with each other. Thus, as shown in Figure 5, the mobile application 40 must pass through the server on each jump between the hosts.

 Figure 6 is a diagram illustrating more details of the mobile application security system 50 shown in Figure 5. In particular, the client/server architecture of the security system in
15 accordance with the invention ensures that the server tracks all of the mobile applications in the system and all of the jumps of all of the mobile applications. The server 52 may also perform security procedures on the mobile applications while they are in transit. Thus, for example, a security check 70 may be performed by the security module of the server each time a mobile
20 application jumps from one host to another host as shown in Figure 6.

The security system 50 in accordance with the invention provides many advantages over the typical mobile application systems. For example, the necessary and feasible security procedures which the server can perform to ensure the security of the mobile application system are provided that raise the level of security of the mobile application system sufficiently to allow deployment in most computer systems. The system may also perform and generate certain responses to a failure of security checks as described below.

In accordance with the invention, since any mobile application must jump to the server between each host, the server may capture and record the entire mobile application during each jump. Then, on subsequent jumps, the server can compare the previously saved mobile application with the new (and potentially changed) mobile application to detect unwanted tampering by each host. The above is just one example of the security checks that can be performed by the server and the server may also perform other security checks as described below. In particular, five different embodiments will be described. Now, a first embodiment of the security system (referred to as "Jumping Beans") will be described that prevents/detects unwanted changes in the mobile application code.

In accordance with the invention, the system may detect unwanted changes in the code of a mobile application and strip unsafe code from mobile applications by a combination of three different processes: 1) never retrieving code from untrusted hosts, (2) preventing untrusted hosts from forwarding code, and (3) marking mobile applications as having immutable code. With Jumping Beans, each participating host can be marked to operate in one of two ways: 1) The host cannot inject any code into the mobile application system, except for code which the host

provides for execution on itself, or 2) All code supplied by the host can be propagated to other hosts in the mobile application system. The hosts are marked this way from the server, so the server is aware of how each host is marked. An example of the implementation of the invention will now be described.

5 Never retrieve code from untrusted hosts

Jumping Beans mobile applications do not necessarily carry with them all of the code needed for execution. Jumping Beans implements a protocol for retrieving any code which the mobile application might require, and this protocol is part of the implementation:

10 a. The mobile application inspects its own internal datastore 47 to see if the required code is available there. If it is, the mobile application uses it and searches no further.

 b. If the mobile application cannot find the requested code in its own datastore, the mobile application queries the local host for the code. The local host inspects its own pre-installed software to determine if the requested code is available there. If it is, the mobile application uses it and searches no further.

15 c. If the mobile application cannot find the requested code, it forms a request for the requested code which is sent to the server.

 d. The server then checks the host from which the mobile application originated. If this host is marked as allowed to inject code into the mobile application system, then the server sends

a request to the originating host for the requested code. If the requested code is found there, the server forwards the code to the mobile application and skips the next step.

e. If the originating host is marked as unable to inject code into the mobile application system, or if the originating host does not have the requested code, then the server inspects its own previously installed software to see if the requested code is available from the server. If it is available from the server, the requested code is forwarded to the mobile application.

f. If the mobile application retrieves the requested code from the server (either from the originating host or from software pre-installed on the server), then the mobile application stores the retrieved code in its own datastore so that it will not need to be retrieved in the future.

g. If the mobile application retrieves the requested code from the server (either from the originating host or from software pre-installed on the server), then the mobile application uses that code and searches no further.

h. If the mobile application cannot retrieve the requested code from the server, then an exception is raised. Figure 7 illustrates an example of the above process.

Figure 7a is a diagram illustrating a first embodiment of the mobile application security system 50 for detecting unwanted changes to the code of a mobile application in accordance with the invention. In particular, the mobile application 40 is created at and resides initially on Host1. In this example, the mobile application 40 is assumed to be marked as having immutable code. Host1 then dispatches the mobile application to Host2. In order to do that, the mobile application is directed to the server 52 that saves a copy of the mobile application's code in the

database. Next, the server forwards the mobile application to the next host (Host2 in this example). At Host2, the mobile application is received, executed and later dispatched to the next host (Host3 in this example). To transfer the mobile application to Host3, the server receives the mobile application again, compares the code of the newly received mobile application with the original code it saved initially to check for various security problems and then, provided that the code has not changed, forwards the mobile application to Host3. The mobile application then arrives at Host3 which executes the mobile application. In summary, on each jump, the server can save the mobile application's code and, on subsequent jumps, the server can compare the previously saved code to the current code of the mobile application in order to ensure that nothing was added to or removed from the code of the mobile application. Now, a second embodiment of the security system in accordance with the invention will be described.

Prevent untrusted hosts from forwarding code.

When a mobile application is dispatched to the server, one of three possible actions is taken:

a. If the host is not allowed to inject code into the system, and the mobile application has never been previously dispatched, then the server simply empties all of the mobile application's code from the mobile application's datastore and saves a copy of the mobile application's empty datastore for future use, and then forwards the mobile application to the next host.

b. If the host is not allowed to inject code into the system, and the mobile application has been dispatched in the past, then the server simply restores the mobile application's datastore to what was saved on the previous jump.

c. If the host is allowed to inject code into the system, then the server inspects the mobile application's ACL, as described next. Figures 8 – 11 illustrate examples of this process.

Figure 8 is a diagram illustrating a first example of a second embodiment of the mobile application security system 50 for preventing hostile hosts from transmitting code to other hosts in accordance with the invention. In particular, the mobile application 40 is created by Host1 and then later dispatched to another host to continue the execution of the mobile application. In this example, Host1 is untrusted in that the server 52 does not know whether or not to trust the host when interacting with the mobile application. Therefore, the mobile application dispatched from Host1 is sent to the server 52 in accordance with the invention and the server may perform several security measures. For example, it may strip any code from the mobile application and store an (empty) copy of the mobile application code in the database 62. The server may alternatively check the code to ensure that it is safe and forward only safe code to the next host. The server may then forward the mobile application onto the next host, Host2 in this example. The mobile application may then be received by and executed by Host2. When the mobile application requires code for execution, the tested version of the code may be supplied to Host2 by the server 52 thus ensuring that the untrusted host cannot spread a virus, for example, using the mobile application. Now, the dispatch of a mobile application from a trusted host to another host will be described.

Figure 9 is a diagram illustrating a second example of a second embodiment of the mobile application security system 50 for preventing hostile hosts from transmitting code to other hosts in accordance with the invention. In particular, the mobile application 40 is created by Host1 and then later dispatched to another host to continue the execution of the mobile application. In this example, Host1 is trusted in that the server 52 knows that the particular host is trusted and therefore does not need to strip the code from the mobile application and test it as described above. Therefore, the mobile application dispatched from Host1 is sent to the server 52 in accordance with the invention and the server may store a copy of the mobile application code in the database 62. The server may then forward the mobile application onto the next host, Host2 in this example. The mobile application may then be received by and executed by Host2. When the mobile application requires the code for execution, the known safe version of the code may be supplied to Host2 by the server 52 or, since the originating host is trusted, the code may be provided by the originating host. Now, the subsequent dispatch of a mobile application from an untrusted host will be described.

Figure 10 is a diagram illustrating a third example of a second embodiment of the mobile application security system 50 for preventing hostile hosts from transmitting code to other hosts in accordance with the invention. In particular, the mobile application 40 is received from another host by an untrusted host (Host n) and then later dispatched to another host to continue the execution of the mobile application. In this example, Host n is untrusted in that the server 52 does not know whether the particular host may perform nefarious acts on the mobile application or using the mobile application. Therefore, the mobile application dispatched from Host n is sent

to the server 52 in accordance with the invention and the server may perform several security measures. For example, the server may receive the code of the mobile application and compare the current code to a previously stored version of the code to ensure that the newly received code is the same as the previous code. The server may then forward the mobile application onto the next host, Host n+1 in this example. The mobile application may then be received by and executed by Host n+1. When the mobile application requires code for execution, the known safe version of the code may be supplied to Host n+1 by the server 52 or, if the originating host is trusted, the code may be provided by the originating host. Now, the subsequent dispatch of a mobile application from a trusted host will be described.

Figure 11 is a diagram illustrating a fourth example of a second embodiment of the mobile application security system 50 for preventing hostile hosts from transmitting code to other hosts in accordance with the invention. In particular, the mobile application 40 is received from another host by a trusted host (Host n) and then later dispatched to another host to continue the execution of the mobile application. In this example, Host n is trusted in that the server 52 knows that the particular host will not perform nefarious acts using the mobile application. Therefore, the mobile application dispatched from Host n is sent to the server 52 in accordance with the invention and the server may perform several security measures. For example, the server may receive the code of the mobile application and store a copy of it in the database 62. No comparison is necessary since the host is trusted. The server may then forward the mobile application onto the next host, Host n+1 in this example. The mobile application may then be received by and executed by Host n+1. When the mobile application requires the code for

execution, the known safe version of the code may be supplied to Host n+1 by the server 52 or, if the originating host is trusted, the code may be provided by the originating host. Now, a third embodiment of the mobile application security system will be described.

Mark mobile applications as having immutable code.

5 The Jumping Beans server may inspect each mobile application's Access Control List (ACL) to determine if the code in that mobile application is immutable. One of three possible tasks actions is taken:

10 a. If the mobile application's code cannot be changed, and the mobile application has never been dispatched in the past, and the mobile application is being dispatched from a trusted host, then the server simply saves the mobile application's code for later use and the mobile application is forwarded to the next host in the itinerary.

15 b. If the mobile application's code cannot be changed, and the mobile application has never been dispatched in the past, and the mobile application is being dispatched from an untrusted host, then the server strips the mobile application's code from the mobile application and saves the mobile application's (empty) code for later use and the mobile application is forwarded to the next host in the itinerary.

 c. If the mobile application's code cannot be changed, and the mobile application has been previously dispatched, then the server discards the mobile application's datastore, and inserts the datastore saved on the previous jump.

d. If the mobile application's code can be changed, then the server simply saves the mobile application's code and forwards the mobile application to the next host without altering its datastore.

Detect unwanted changes in the mobile application's state

5 The Jumping Beans server inspects each mobile application's Access Control List (ACL) to determine if the state of that mobile application is immutable. One of three possible actions is taken:

10 a. If the mobile application's state cannot be changed, and the mobile application has never been dispatched in the past, then the server saves the mobile application's state for later use and the mobile application is forwarded to the next host in the itinerary;

 b. If the mobile application's code cannot be changed, and the mobile application has been previously dispatched, then the server discards the mobile application's state, and inserts the state saved on the previous jump.

15 c. If the mobile application's code can be changed, then the server simply saves the mobile application's state for later use and the mobile application is forwarded to the next host in the itinerary. Figure 12 illustrates an example of the process.

Figure 12 is a diagram illustrating a third embodiment of the mobile application security system 50 for detecting unwanted changes to the state of a mobile application in accordance with the invention. In general, the server 52 may compare the state of the mobile application on the

previous jump with the state of the mobile application on the current jump. This allows the server to detect the unwanted changes in the state of the mobile application. In more detail, a host, Host1 in this example, may create a mobile application 40 that is then dispatched to other hosts for further execution. When the mobile application 40 is dispatched, it is sent to the server 5 52 which may save a copy of the mobile application's state. The server may then forward the mobile application to the next host, Host2 in this example. Host2 may receive the mobile application, execute it and then forward it onto the next host. The server may receive the mobile application from the next host and compare the state of the mobile application received from the next host to the state of the mobile application saved in the database to determine if changes have 10 occurred. If the comparison does not detect any unwanted changes with the mobile application, the server may forward the mobile application onto the next host. Thus, in this embodiment, a host that executes the mobile application is unable to insert changes into the mobile application's state since those changes will be identified by the server when the comparison step is executed by the server. Now, a fourth embodiment of the mobile application security system will be 15 described.

Enforcing a mobile application's itinerary

The ACL in a mobile application can indicate whether or not that mobile application's itinerary can be edited. Even if a mobile application's ACL indicates that the mobile application's itinerary can be edited, under no circumstances should that portion of an itinerary 20 which represents the previous history of the mobile application ever be altered, nor should it ever be inaccurate. Because each mobile application must pass through the server on each jump, the

server can accurately track the current and past locations of each mobile application. On a mobile application's first jump, the server simply saves that mobile application's entire itinerary for later use, and then forwards the mobile application to the next host. On subsequent jumps, the server inspects the mobile application's ACL, and handles the mobile application's itinerary in one of

5 two ways:

a. If the mobile application's itinerary can be edited, the server simply ensures that the past itinerary accurately reflects the mobile application's past visits. If the mobile application's past itinerary does not match the server's record, a security exception is thrown.

b. If the mobile application's itinerary can not be edited, the server compares the mobile

10 application's entire itinerary to the itinerary saved on the previous jump. If there is any difference, a security exception is thrown. On every jump, the server saves each mobile application's entire itinerary for later use. Figures 13 – 15 illustrate examples of this process.

Figure 13 is a diagram illustrating a first example of a fourth embodiment of the mobile application security system 50 for detecting unwanted changes in the itinerary of the mobile

15 application in accordance with the invention. In general, on each jump of the mobile application, the server may determine the host from which the mobile application was dispatched and the hosts to which the mobile application is dispatched. In particular, this permits the server 52 to enforce the itinerary (e.g., the hosts where the mobile application is going to be executed) of the mobile application. In more detail, a first host (Host1) may create a mobile application 40 and

20 then may dispatch the mobile application to another host through the server 52 in accordance with the invention. When the server receives the mobile application 40, the server 52 may store

a copy of the itinerary of the mobile application in the database 62. The server may then forward the mobile application to the next host (Host2) according to the itinerary. Now, another example of the embodiment for detecting changes in the itinerary will be described.

Figure 14 is a diagram illustrating a second example of a fourth embodiment of the mobile application security system 50 for detecting unwanted changes in the itinerary of the mobile application in accordance with the invention wherein the itinerary of a mobile application is already stored in the server. In more detail, a first host (Host n) may dispatch a mobile application 40 to another host through the server 52 in accordance with the invention. When the server receives the mobile application 40, the server 52 may compare the current itinerary of the mobile application to a stored copy of the itinerary to ensure they match each other. If the itineraries match, then the server may forward the mobile application onto the next host (Host n+1) that receives the mobile application and executes it. Now, another example of the embodiment for detecting changes in the itinerary will be described.

Figure 15 is a diagram illustrating a third example of a fourth embodiment of the mobile application security system 50 for detecting unwanted changes in the itinerary of the mobile application in accordance with the invention wherein the itinerary may be changed. In more detail, a first host (Host n) which has received a mobile application 40 from another host may dispatch the mobile application. The mobile application then passes through the server 52 in accordance with the invention. When the server receives the mobile application in accordance with the invention, it may ensure that the historical portion of the itinerary is accurate by comparing the previously saved itinerary with the new itinerary. If the historical portion of the

itinerary is accurate, the server forwards the mobile application to the next host (Host n+1).

Now, a fifth embodiment of the mobile application security system will be described.

Figure 16 is a diagram illustrating a first example of a fifth embodiment of the mobile application security system 50 for preventing untrusted hosts from launching a mobile application in accordance with the invention. In general, on each jump of the mobile application, the server may determine if the mobile application has previously been in the system. For example, if the host from which the mobile application is sent is an untrusted host, the server may prevent the mobile application from being forwarded to the next host. In more detail, as shown in Figure 16, a first host (Host1) may create a mobile application 40 and then later dispatch it to another host. In accordance with the invention, the dispatched mobile application first is sent to the server 52. The server 52 may determine that the mobile application is new and therefore further investigation is necessary. If the server then determines that the particular host is allowed (e.g., is trusted to) to launch mobile applications, the server may forward the mobile application to the next host (Host2) so that Host2 receives the mobile application.

Figure 17 is a diagram illustrating a second example of a fifth embodiment of the mobile application security system 50 for preventing untrusted hosts from launching a mobile application in accordance with the invention. In particular, an untrusted host (Host1) may create a new mobile application that is then later dispatched. The mobile application is then sent dispatched to the server 52 first in accordance with the invention. The server 52 determines that the host dispatching the mobile application is untrusted so that the server does not forward the mobile application to the next host.

Figure 18 is a diagram illustrating a third example of a fifth embodiment of the mobile application security system 50 for preventing untrusted hosts from launching a mobile application in accordance with the invention wherein a subsequent dispatch of the mobile application occurs. In particular, a host (Host n) attempts to dispatch a mobile application to another host which must pass through the server 52 in accordance with the invention. When the mobile application is received by the server, the server may determine that the mobile application is not new (e.g., the server knows about the mobile application and knows that it is safe) and forwards the mobile application to the next host (Host n+1). Now, a summary of how the above procedures raise the security level of a mobile application environment will be described.

The most serious security problem perceived by industry observers is that a mobile application system allows a hostile host to inject dangerous code into a computing system, and there is no way to detect this. By marking a host so that it is not allowed to inject code into the system, the other hosts in the mobile application system do not have to trust any code originating from that host. Instead, they only need to trust the server in accordance with the invention.

Another security problem often cited by industry observers is that an hostile host can modify the code of the mobile application to give it undesirable behavior, then forward the mobile application other hosts in the system. Most (but not all) mobile applications, as deployed in real-world systems, will have fixed code, meaning that the code will not change during the lifetime of the mobile application. Virtually all mobile applications can be designed so that they do not require that the code change. On creation, a mobile application's ACL can be set up so that its code cannot be altered in accordance with the invention. This prevents an hostile host

from modifying a mobile application's code and forwarding that modified code to other hosts. A few (but not many) mobile applications will not need to alter their state during their life-time. When creating the mobile application, the ACL can be set up so that its state cannot be altered in accordance with the invention.

5 Another security concern often cited by industry observers is that an hostile host can tamper with a mobile application in an unwanted way, and then forward that contaminated mobile application to other hosts. This problem is a superset of the problem above. As described above, the security technology described in this can protect a mobile application's code. The two remaining major pieces of a mobile application are its state and its itinerary. As described
10 elsewhere in this document, a mobile application's itinerary can be protected from an hostile host. The only possible remaining method of attack by a hostile host is to alter the mobile application's state. Once a mobile application's code and itinerary are protected, the problem is reduced to the exact same problem faced by distributed computing systems which don't use mobility. Systems which don't use mobility are passing around simple data. As this data is
15 passed around, the pre-installed software on the different computers will respond to, alter, and otherwise process this data. The state of a mobile application is just data, exactly the same as the data passed around in traditional computing systems. Basically, a mobile application system can be secured by applying the technology described herein. Now, possible responses by the mobile application security system to security violations will be described.

20 In one embodiment, the server could accept the mobile application from the sending host and then destroy the mobile application. In another embodiment, the server could perform the

security procedures before acknowledging receipt of the mobile application. If the security procedures fail, the server could reject the mobile application and leave it on the offending host.

In yet another embodiment, the server could correct the violation, and then forward the mobile application to the next host although this is not possible for all types of security violations. In all

5 cases where the security procedures fail, the server should record such events in the audit logs.

While the foregoing has been with reference to a particular embodiment of the invention, it will be appreciated by those skilled in the art that changes in this embodiment may be made without departing from the principles and spirit of the invention, the scope of which is defined by the appended claims.

Claims:

1 1. A mobile application security system, comprising:
2 a central computer for controlling the security of a mobile application;
3 one or more host computers connected to the server computer, each host computer
4 executing the mobile application that jumps between the hosts during execution;
5 the central computer further comprising means for monitoring the security of the mobile
6 application as it jumps between the host computers wherein when the mobile application is
7 communicated from a first host to a second host, it passes through the central computer; and
8 wherein the security monitoring means further comprises means for detecting unwanted
9 changes in the code associated with the mobile application when the mobile application is
10 jumping between hosts.

1 2. The system of Claim 1, wherein the detecting means further comprises means for
2 storing a copy of the mobile application when the mobile application first passes through the
3 server, means for receiving the mobile application after it is executed by another host and means
4 for comparing the code of the mobile application after it is executed by another host to the stored
5 copy of the mobile application to determine if changes have been made to the code of the mobile
6 application.

1 3. A mobile application security system, comprising:
2 a central computer for controlling the security of a mobile application;
3 one or more host computers connected to the server computer, each host computer
4 executing the mobile application that jumps between the hosts during execution;

5 the central computer further comprising means for monitoring the security of the mobile
6 application as it jumps between the host computers wherein when the mobile application is
7 communicated from a first host to a second host, it passes through the central computer; and
8 wherein the security monitoring means further comprises means for preventing a host
9 from transmitting hostile code in a mobile application to another host.

1 4. The system of Claim 3, wherein the preventing means further comprises means
2 for determining if the host dispatching the mobile application is trusted, means for stripping the
3 code from an initially received mobile application if the host is not trusted, means for saving the
4 code of the mobile application, and means, when requested by another host, for providing the
5 code for the mobile application to the requesting host.

1 5. A mobile application security system, comprising:
2 a central computer for controlling the security of a mobile application;
3 one or more host computers connected to the server computer, each host computer
4 executing the mobile application that jumps between the hosts during execution;
5 the central computer further comprising means for monitoring the security of the mobile
6 application as it jumps between the host computers wherein when the mobile application is
7 communicated from a first host to a second host, it passes through the central computer; and
8 wherein security monitoring means further comprises means for detecting unwanted
9 changes in the state of the mobile application.

1 6. The system of Claim 5, wherein the detecting means further comprises means for
2 saving a copy of the state of a received mobile application, means for receiving the same mobile
3 application after a jump to another host and means for comparing the state of the mobile

4 application after the jump to another host with the stored state of the mobile application to ensure
5 that the state of the mobile application has not changed.

1 7. A mobile application security system, comprising:
2 a central computer for controlling the security of a mobile application;
3 one or more host computers connected to the server computer, each host computer
4 executing the mobile application that jumps between the hosts during execution;
5 the central computer further comprising means for monitoring the security of the mobile
6 application as it jumps between the host computers wherein when the mobile application is
7 communicated from a first host to a second host, it passes through the central computer; and
8 wherein the security monitoring means further comprises means for detecting unwanted
9 changes in the itinerary of the mobile application.

1 8. The system of Claim 7, wherein the detecting means further comprises means for
2 saving a copy of the itinerary of a received mobile application, means for receiving the same
3 mobile application after a jump to another host and means for comparing the itinerary of the
4 mobile application after the jump to another host with the stored itinerary of the mobile
5 application to ensure that the itinerary of the mobile application has not changed.

1 9. The system of Claim 7, wherein the itinerary comprises past historical itinerary
2 data.

1 10. A mobile application security method, comprising:
2 receiving a mobile application at a central computer each time the mobile application is
3 jumping between a first host and a second host; and

4 monitoring the security of the mobile application as it jumps between the host computers,
5 wherein the security monitoring further comprises detecting unwanted changes in the code
6 associated with the mobile application when the mobile application is jumping between hosts.

1 11. The method of Claim 10, wherein the detecting further comprises storing a copy
2 of the mobile application when the mobile application first passes through the server, receiving
3 the mobile application after it is executed by another host and comparing the code of the mobile
4 application after it is executed by another host to the stored copy of the mobile application to
5 determine if changes have been made to the code of the mobile application.

1 12. A mobile application security method, comprising:
2 receiving a mobile application at a central computer each time the mobile application is
3 jumping between a first host and a second host; and
4 monitoring the security of the mobile application as it jumps between the host computers,
5 wherein the security monitoring further comprises preventing a host from transmitting hostile
6 code in a mobile application to another host.

1 13. The method of Claim 12, wherein the preventing further comprises determining if
2 the host dispatching the mobile application is trusted, stripping the code from an initially
3 received mobile application if the host is not trusted, saving the code of the mobile application,
4 and, when requested by another host, providing the code for the mobile application to the
5 requesting host.

1 14. A mobile application security method, comprising:
2 receiving a mobile application at a central computer each time the mobile application is
3 jumping between a first host and a second host; and

4 monitoring the security of the mobile application as it jumps between the host computers,
5 wherein the security monitoring further comprises detecting unwanted changes in the
6 state of the mobile application.

1 15. The method of Claim 14, wherein the detecting further comprises saving a copy of
2 the state of a received mobile application, receiving the same mobile application after a jump to
3 another host and comparing the state of the mobile application after the jump to another host
4 with the stored state of the mobile application to ensure that the state of the mobile application
5 has not changed.

1 16. A mobile application security method, comprising:
2 receiving a mobile application at a central computer each time the mobile application is
3 jumping between a first host and a second host; and
4 monitoring the security of the mobile application as it jumps between the host computers,
5 wherein the security monitoring further comprises detecting unwanted changes in the itinerary of
6 the mobile application.

1 17. The method of Claim 16, wherein the detecting further comprises saving a copy of
2 the itinerary of a received mobile application, receiving the same mobile application after a jump
3 to another host and comparing the itinerary of the mobile application after the jump to another
4 host with the stored itinerary of the mobile application to ensure that the itinerary of the mobile
5 application has not changed.

1 18. The method of Claim 16, wherein the itinerary comprises past historical itinerary
2 data.

1 19. A mobile application security method, comprising:

- 2 receiving a mobile application at a central computer each time the mobile application is
3 jumping between a first host and a second host; and
4 monitoring the security of the mobile application as it jumps between the host computers,
5 wherein the security monitoring further comprises preventing untrusted hosts from initially
6 launching mobile applications

03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 1042 1043 1044 1045 1046 1047 1048 1049 1050 1051 1052 1053 1054 1055 1056 1057 1058 1059 1060 1061 1062 1063 1064 1065 1066 1067 1068 1069 1070 1071 1072 1073 1074 1075 1076 1077 1078 1079 1080 1081 1082 1083 1084 1085 1086 1087 1088 1089 1090 1091 1092 1093 1094 1095 1096 1097 1098 1099 1100 1101 1102 1103 1104 1105 1106 1107 1108 1109 1110 1111 1112 1113 1114 1115 1116 1117 1118 1119 1120 1121 1122 1123 1124 1125 1126 1127 1128 1129 1130 1131 1132 1133 1134 1135 1136 1137 1138 1139 1140 1141 1142 1143 1144 1145 1146 1147 1148 1149 1150 1151 1152 1153 1154 1155 1156 1157 1158 1159 1160 1161 1162 1163 1164 1165 1166 1167 1168 1169 1170 1171 1172 1173 1174 1175 1176 1177 1178 1179 1180 1181 1182 1183 1184 1185 1186 1187 1188 1189 1190 1191 1192 1193 1194 1195 1196 1197 1198 1199 1200 1201 1202 1203 1204 1205 1206 1207 1208 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 1220 1221 1222 1223 1224 1225 1226 1227 1228 1229 1230 1231 1232 1233 1234 1235 1236 1237 1238 1239 1240 1241 1242 1243 1244 1245 1246 1247 1248 1249 1250 1251 1252 1253 1254 1255 1256 1257 1258 1259 1260 1261 1262 1263 1264 1265 1266 1267 1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280 1281 1282 1283 1284 1285 1286 1287 1288 1289 1290 1291 1292 1293 1294 1295 1296 1297 1298 1299 1300 1301 1302 1303 1304 1305 1306 1307 1308 1309 1310 1311 1312 1313 1314 1315 1316 1317 1318 1319 1320 1321 1322 1323 1324 1325 1326 1327 1328 1329 1330 1331 1332 1333 1334 1335 1336 1337 1338 1339 1340 1341 1342 1343 1344 1345 1346 1347 1348 1349 1350 1351 1352 1353 1354 1355 1356 1357 1358 1359 1360 1361 1362 1363 1364 1365 1366 1367 1368 1369 1370 1371 1372 1373 1374 1375 1376 1377 1378 1379 1380 1381 1382 1383 1384 1385 1386 1387 1388 1389 1390 1391 1392 1393 1394 1395 1396 1397 1398 1399 1400 1401 1402 1403 1404 1405 1406 1407 1408 1409 1410 1411 1412 1413 1414 1415 1416 1417 1418 1419 1420 1421 1422 1423 1424 1425 1426 1427 1428 1429 1430 1431 1432 1433 1434 1435 1436 1437 1438 1439 1440 1441 1442 1443 1444 1445 1446 1447 1448 1449 1450 1451 1452 1453 1454 1455 1456 1457 1458 1459 1460 1461 1462 1463 1464 1465 1466 1467 1468 1469 1470 1471 1472 1473 1474 1475 1476 1477 1478 1479 1480 1481 1482 1483 1484 1485 1486 1487 1488 1489 1490 1491 1492 1493 1494 1495 1496 1497 1498 1499 1500 1501 1502 1503 1504 1505 1506 1507 1508 1509 1510 1511 1512 1513 1514 1515 1516 1517 1518 1519 1520 1521 1522 1523 1524 1525 1526 1527 1528 1529 1530 1531 1532 1533 1534 1535 1536 1537 1538 1539 1540 1541 1542 1543 1544 1545 1546 1547 1548 1549 1550 1551 1552 1553 1554 1555 1556 1557 1558 1559 1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570 1571 1572 1573 1574 1575 1576 1577 1578 1579 1580 1581 1582 1583 1584 1585 1586 1587 1588 1589 1590 1591 1592 1593 1594 1595 1596 1597 1598 1599 1600 1601 1602 1603 1604 1605 1606 1607 1608 1609 1610 1611 1612 1613 1614 1615 1616 1617 1618 1619 1620 1621 1622 1623 1624 1625 1626 1627 1628 1629 1630 1631 1632 1633 1634 1635 1636 1637 1638 1639 1640 1641 1642 1643 1644 1645 1646 1647 1648 1649 1650 1651 1652 1653 1654 1655 1656 1657 1658 1659 1660 1661 1662 1663 1664 1665 1666 1667 1668 1669 1670 1671 1672 1673 1674 1675 1676 1677 1678 1679 1680 1681 1682 1683 1684 1685 1686 1687 1688 1689 1690 1691 1692 1693 1694 1695 1696 1697 1698 1699 1700 1701 1702 1703 1704 1705 1706 1707 1708 1709 1710 1711 1712 1713 1714 1715 1716 1717 1718 1719 1720 1721 1722 1723 1724 1725 1726 1727 1728 1729 1730 1731 1732 1733 1734 1735 1736 1737 1738 1739 1740 1741 1742 1743 1744 1745 1746 1747 1748 1749 1750 1751 1752 1753 1754 1755 1756 1757 1758 1759 1760 1761 1762 1763 1764 1765 1766 1767 1768 1769 1770 1771 1772 1773 1774 1775 1776 1777 1778 1779 1780 1781 1782 1783 1784 1785 1786 1787 1788 1789 1790 1791 1792 1793 1794 1795 1796 1797 1798 1799 1800 1801 1802 1803 1804 1805 1806 1807 1808 1809 1810 1811 1812 1813 1814 1815 1816 1817 1818 1819 1820 1821 1822 1823 1824 1825 1826 1827 1828 1829 1830 1831 1832 1833 1834 1835 1836 1837 1838 1839 1840 1841 1842 1843 1844 1845 1846 1847 1848 1849 1850 1851 1852 1853 1854 1855 1856 1857 1858 1859 1860 1861 1862 1863 1864 1865 1866 1867 1868 1869 1870 1871 1872 1873 1874 1875 1876 1877 1878 1879 1880 1881 1882 1883 1884 1885 1886 1887 1888 1889 1890 1891 1892 1893 1894 1895 1896 1897 1898 1899 1900 1901 1902 1903 1904 1905 1906 1907 1908 1909 1910 1911 1912 1913 1914 1915 1916 1917 1918 1919 1920 1921 1922 1923 1924 1925 1926 1927 1928 1929 1930 1931 1932 1933 1934 1935 1936 1937 1938 1939 1940 1941 1942 1943 1944 1945 1946 1947 1948 1949 1950 1951 1952 1953 1954 1955 1956 1957 1958 1959 1960 1961 1962 1963 1964 1965 1966 1967 1968 1969 1970 1971 1972 1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024 2025 2026 2027 2028 2029 2030 2031 2032 2033 2034 2035 2036 2037 2038 2039 2040 2041 2042 2043 2044 2045 2046 2047 2048 2049 2050 2051 2052 2053 2054 2055 2056 2057 2058 2059 2060 2061 2062 2063 2064 2065 2066 2067 2068 2069 2070 2071 2072 2073 2074 2075 2076 2077 2078 2079 2080 2081 2082 2083 2084 2085 2086 2087 2088 2089 2090 2091 2092 2093 2094 2095 2096 2097 2098 2099 2100 2101 2102 2103 2104 2105 2106 2107 2108 2109 2110 2111 2112 2113 2114 2115 2116 2117 2118 2119 2120 2121 2122 2123 2124 2125 2126 2127 2128 2129 2130 2131 2132 2133 2134 2135 2136 2137 2138 2139 2140 2141 2142 2143 2144 2145 2146 2147 2148 2149 2150 2151 2152 2153 2154 2155 2156 2157 2158 2159 2160 2161 2162 2163 2164 2165 2166 2167 2168 2169 2170 2171 2172 2173 2174 2175 2176 2177 2178 2179 2180 2181 2182 2183 2184 2185 2186 2187 2188 2189 2190 2191 2192 2193 2194 2195 2196 2197 2198 2199 2200 2201 2202 2203 2204 2205 2206 2207 2208 2209 2210 2211 2212 2213 2214 2215 2216 2217 2218 2219 2220 2221 2222 2223 2224 2225 2226 2227 2228 2229 2230 2231 2232 2233 2234 2235 2236 2237 2238 2239 2240 2241 2242 2243 2244 2245 2246 2247 2248 2249 2250 2251 2252 2253 2254 2255 2256 2257 2258 2259 2260 2261 2262 2263 2264 2265 2266 2267 2268 2269 2270 2271 2272 2273 2274 2275 2276 2277 2278 2279 2280 2281 2282 2283 2284 2285 2286 2287 2288 2289 2290 2291 2292 2293 2294 2295 2296 2297 2298 2299 2300 2301 2302 2303 2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315 2316 2317 2318 2319 2320 2321 2322 2323 2324 2325 2326 2327 2328 2329 2330 2331 2332 2333 2334 2335 2336 2337 2338 2339 2340 2341 2342 2343 2344 2345 2346 2347 2348 2349 2350 2351 2352 2353 2354 2355 2356 2357 2358 2359 2360 2361 2362 2363 2364 2365 2366 2367 2368 2369 2370 2371 2372 2373 2374 2375 2376 2377 2378 2379 2380 2381 2382 2383 2384 2385 2386 2387 2388 2389 2390 2391 2392 2393 2394 2395 2396 2397 2398 2399 2400 2401 2402 2403 2404 2405 2406 2407 2408 2409 2410 2411 2412 2413 2414 2415 2416 2417 2418 2419 2420 2421 2422 2423 2424 2425 2426 2427 2428 2429 2430 2431 2432 2433 2434 2435 2436 2437 2438 2439 2440 2441 2442 2443 2444 2445 2446 2447 2448 2449 2450 2451 2452 2453 2454 2455 2456 2457 2458 2459 2460 2461 2462 2463 2464 2465 2466 2467 2468 2469 2470 2471 2472 2473 2474 2475 2476 2477 2478 2479 2480 2481 2482 2483 2484 2485 2486 2487 2488 2489 2490 2491 2492 2493 2494 2495 2496 2497 2498 2499 2500 2501 2502 2503 2504 2505 2506 2507 2508 2509 2510 2511 2512 2513 2514 2515 2516 2517 2518 2519 2520 2521 2522 2523 2524 2525 2526 2527 2528 2529 2530 2531 2532 2533 2534 2535 2536 2537 2538 2539 2540 2541 2542 2543 2544 2545 2546 2547 2548 2549 2550 2551 2552 2553 2554 2555 2556 2557 2558 2559 2560 2561 2562 2563 2564 2565 2566 2567 2568 2569 2570 2571 2572 2573 2574 2575 2576 2577 2578 2579 2580 2581 2582 2583 2584 2585 2586 2587 2588 2589 2590 2591 2592 2593 2594 2595 2596 2597 2598 2599 2600 2601 2602 2603 2604 2605 2606 2607 2608 2609 2610 2611 2612 2613 2614 2615 2616 2617 2618 2619 2620 2621 2622 2623 2624 2625 2626 2627 2628 2629 2630 2631 2632 2633 2634 2635 2636 2637 2638 2639 2640 2641 2642 2643 2644 2645 2646 2647 2648 2649 2650 2651 2652 2653 2654 2655 2656 2657 2

ABSTRACT OF THE DISCLOSURE

The mobile application security system and method in accordance with the invention increases the overall level of security in using a mobile application. In a preferred embodiment, the system may use a client/server architecture wherein each host of a mobile application is treated as a client and a central computer is treated as the server. In operation, any time that a mobile application is going to jump between hosts, it must first pass through the central computer so that the central computer may perform various security checks. The security checks ensure that the security of the mobile application is not compromised and overcomes the above problems with typical mobile application systems.

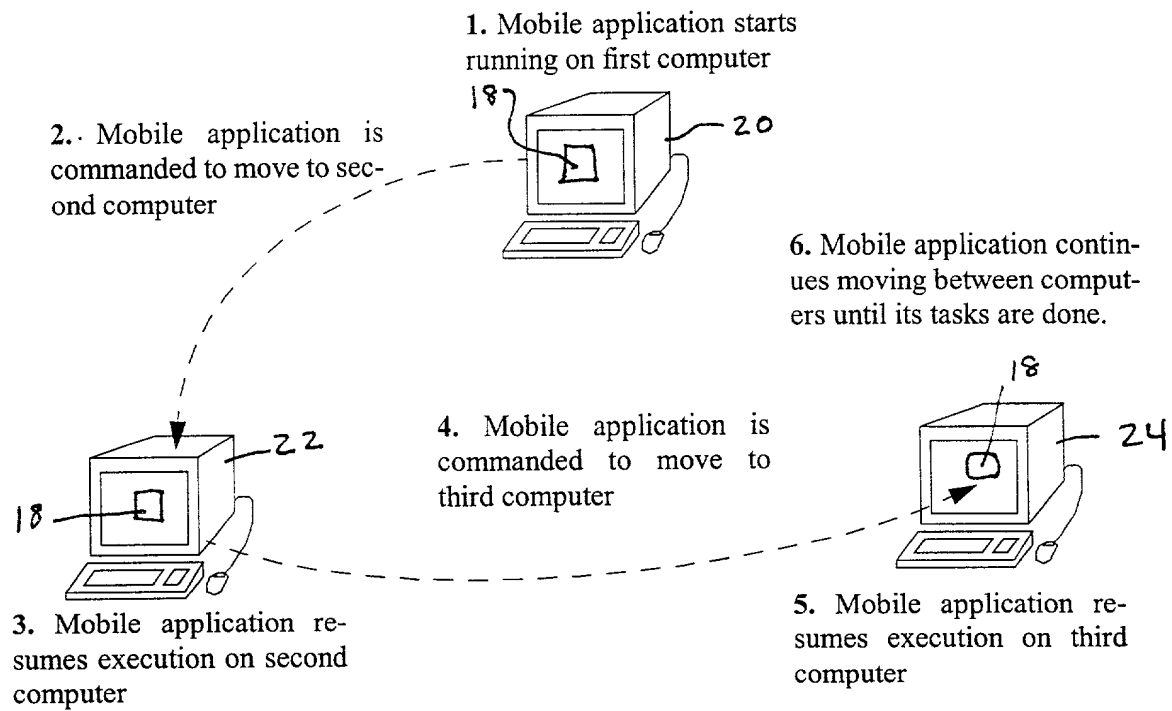


Figure 2

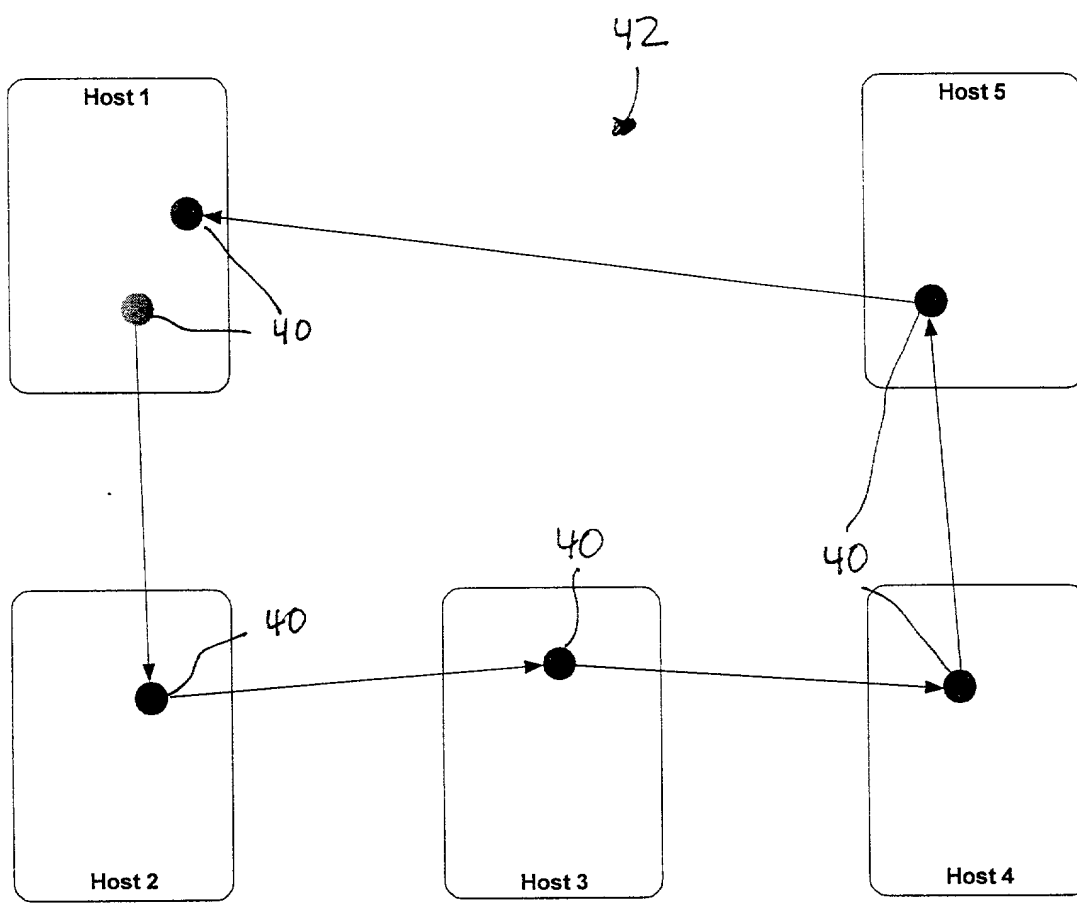


Figure 3

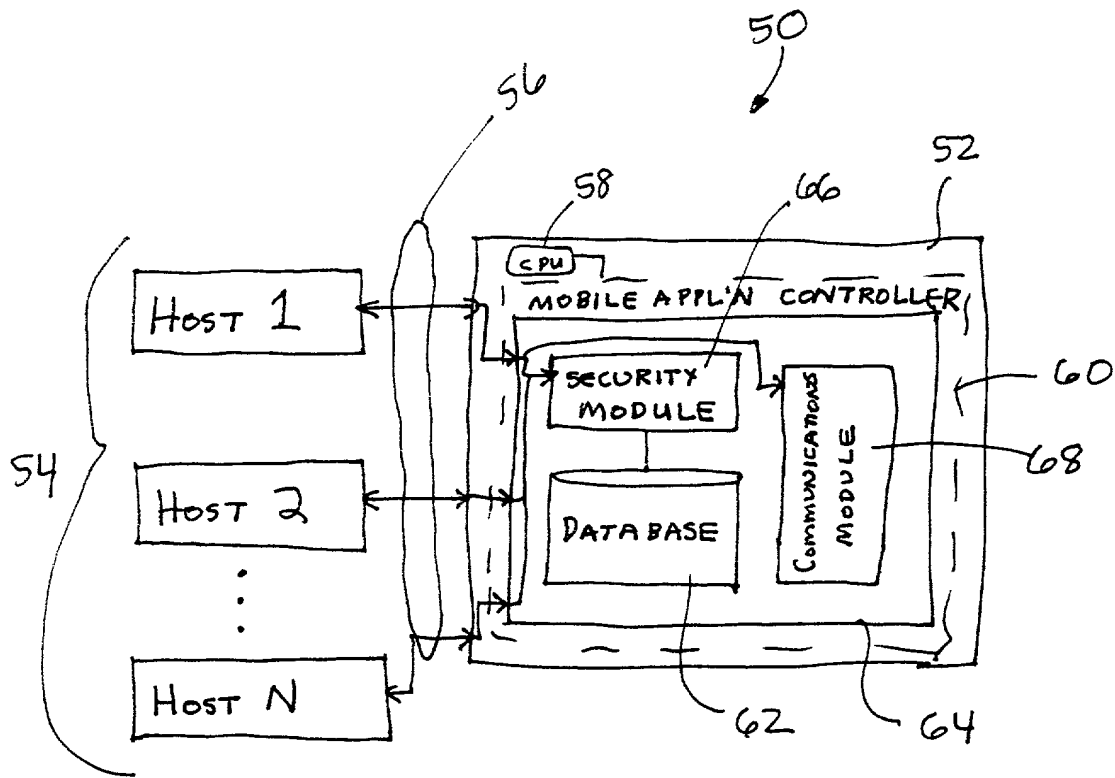


FIGURE 4

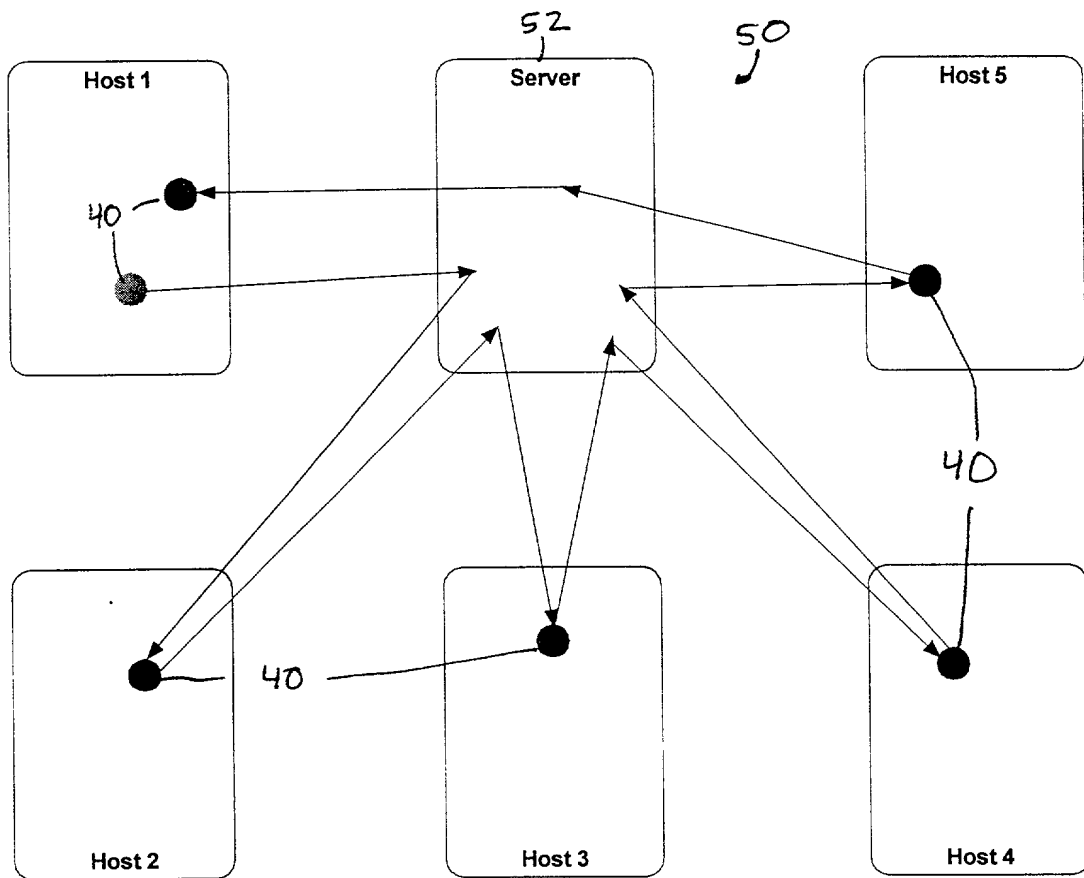


Figure 5

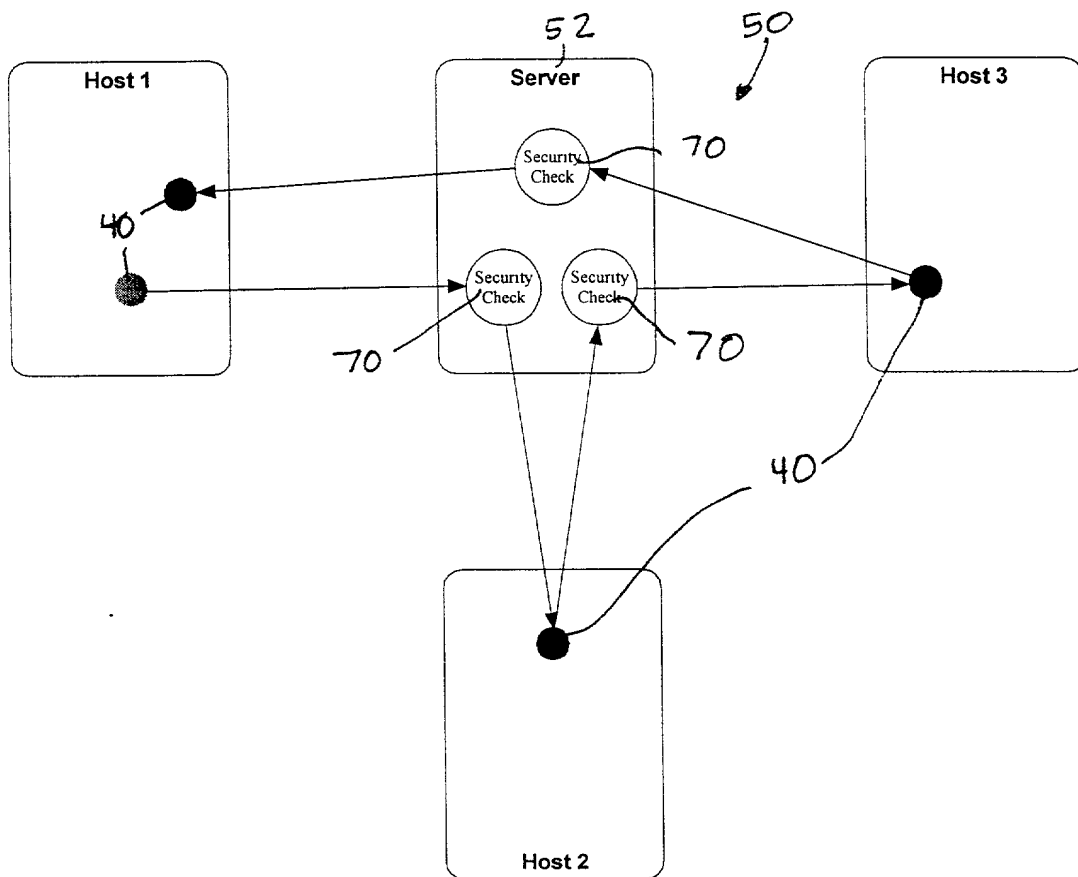


Figure 6

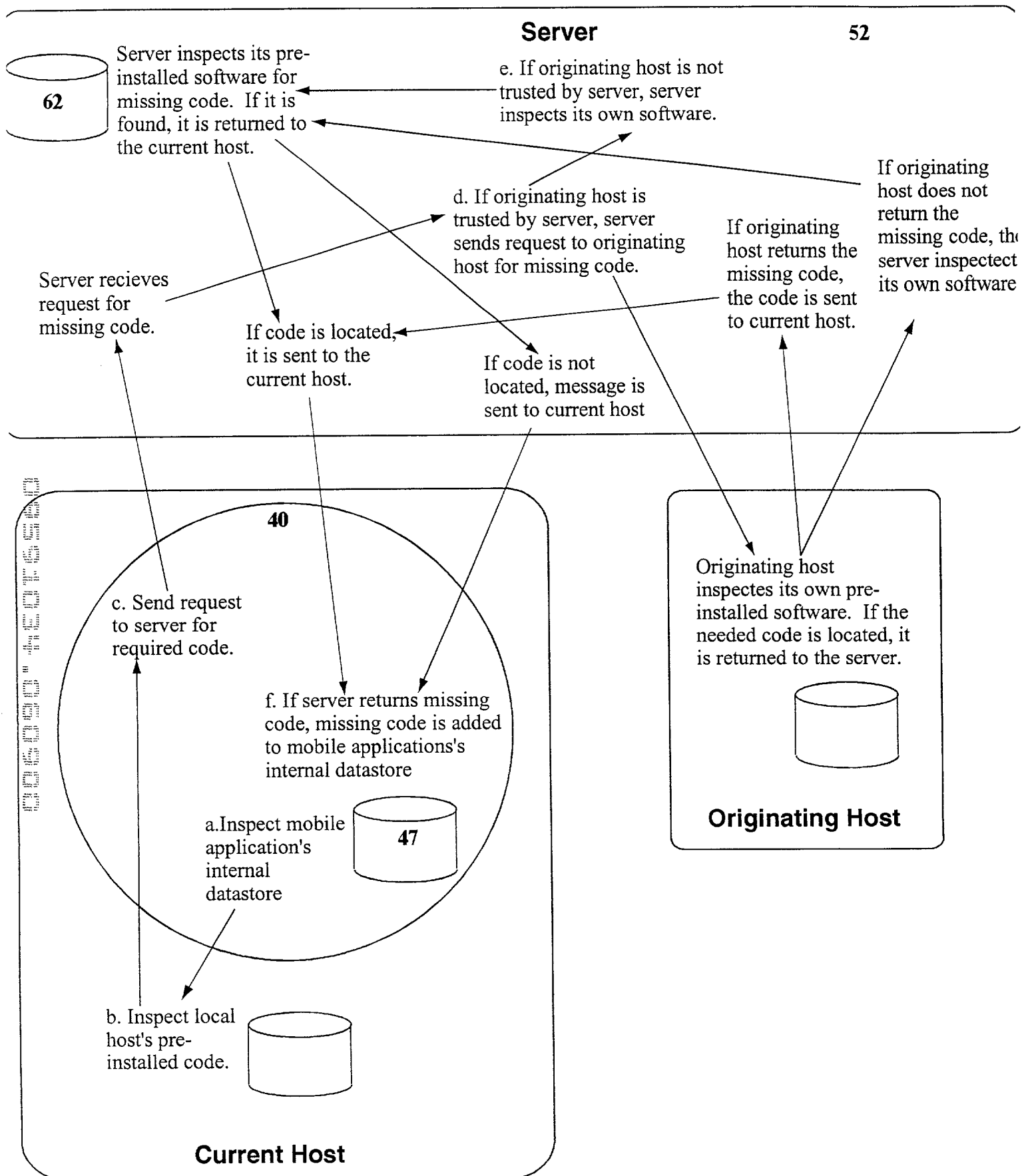


Figure 7

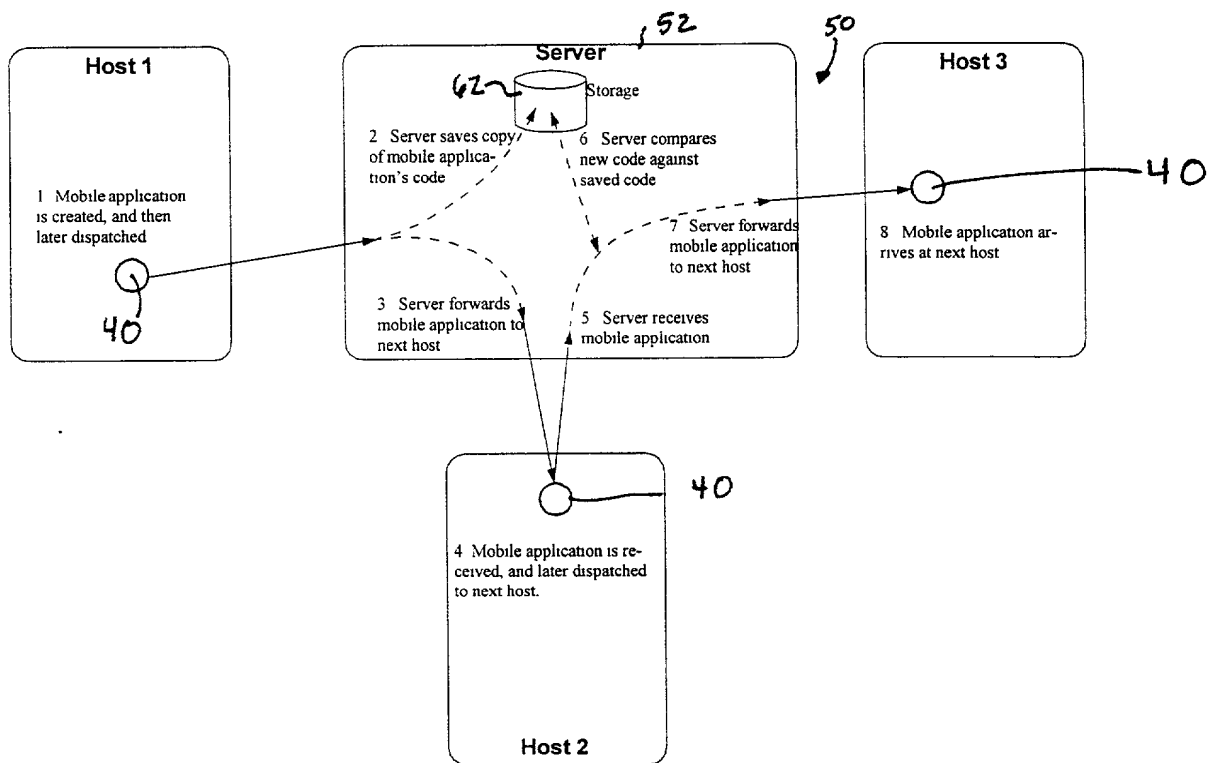
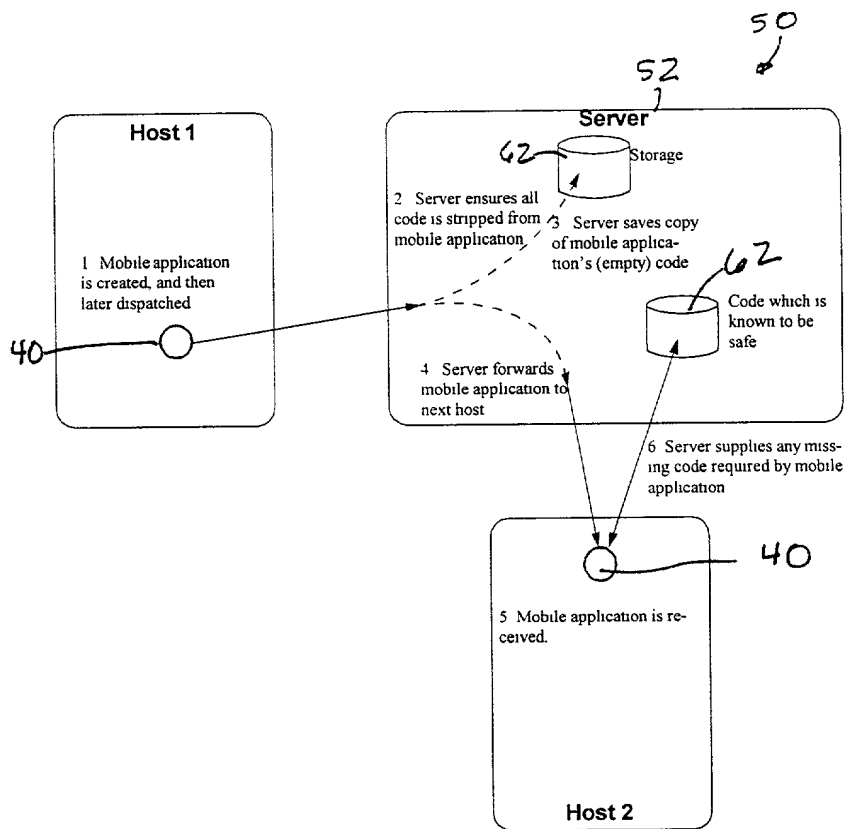


Figure 7a



11.1.3

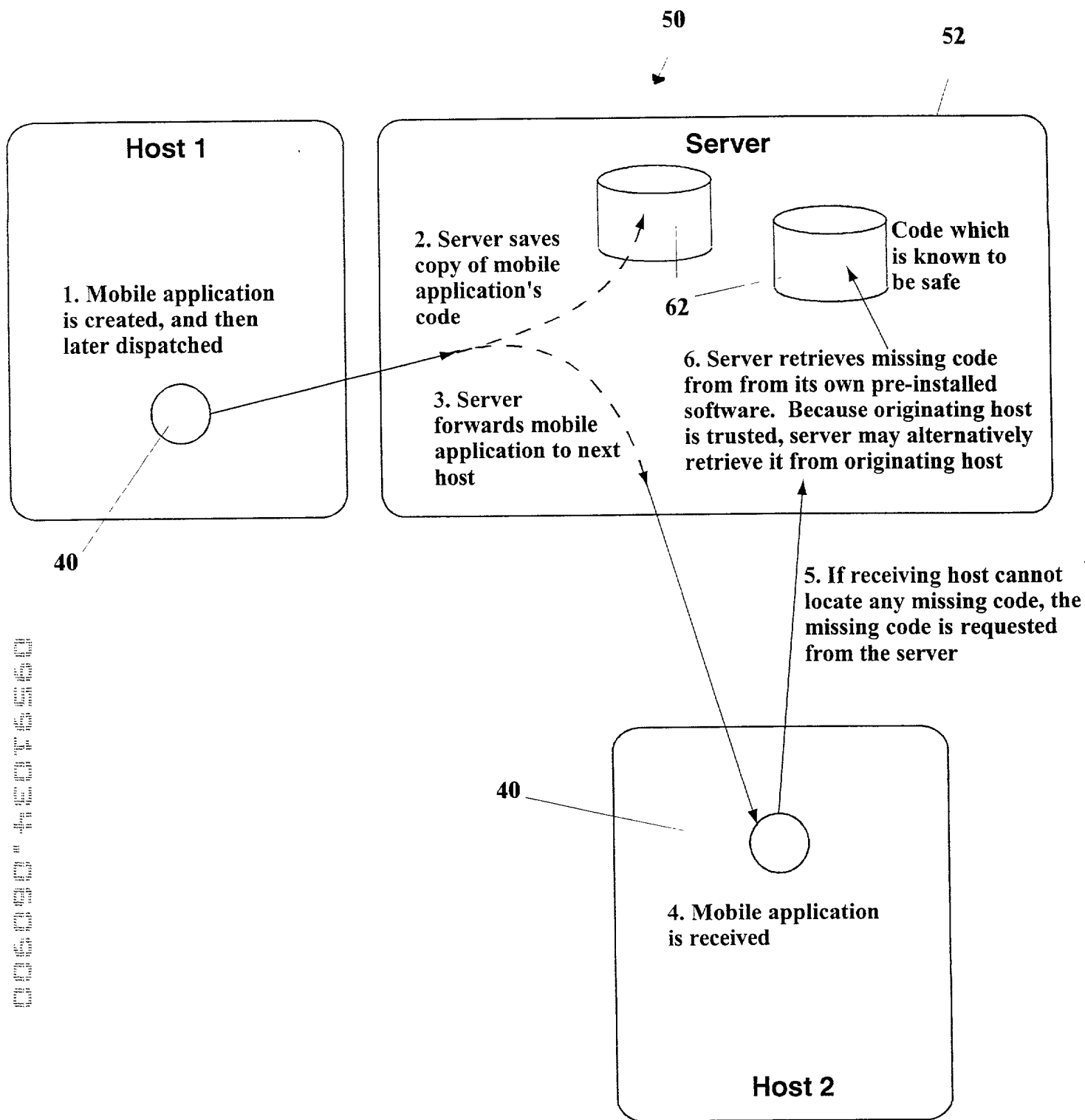


Figure 9

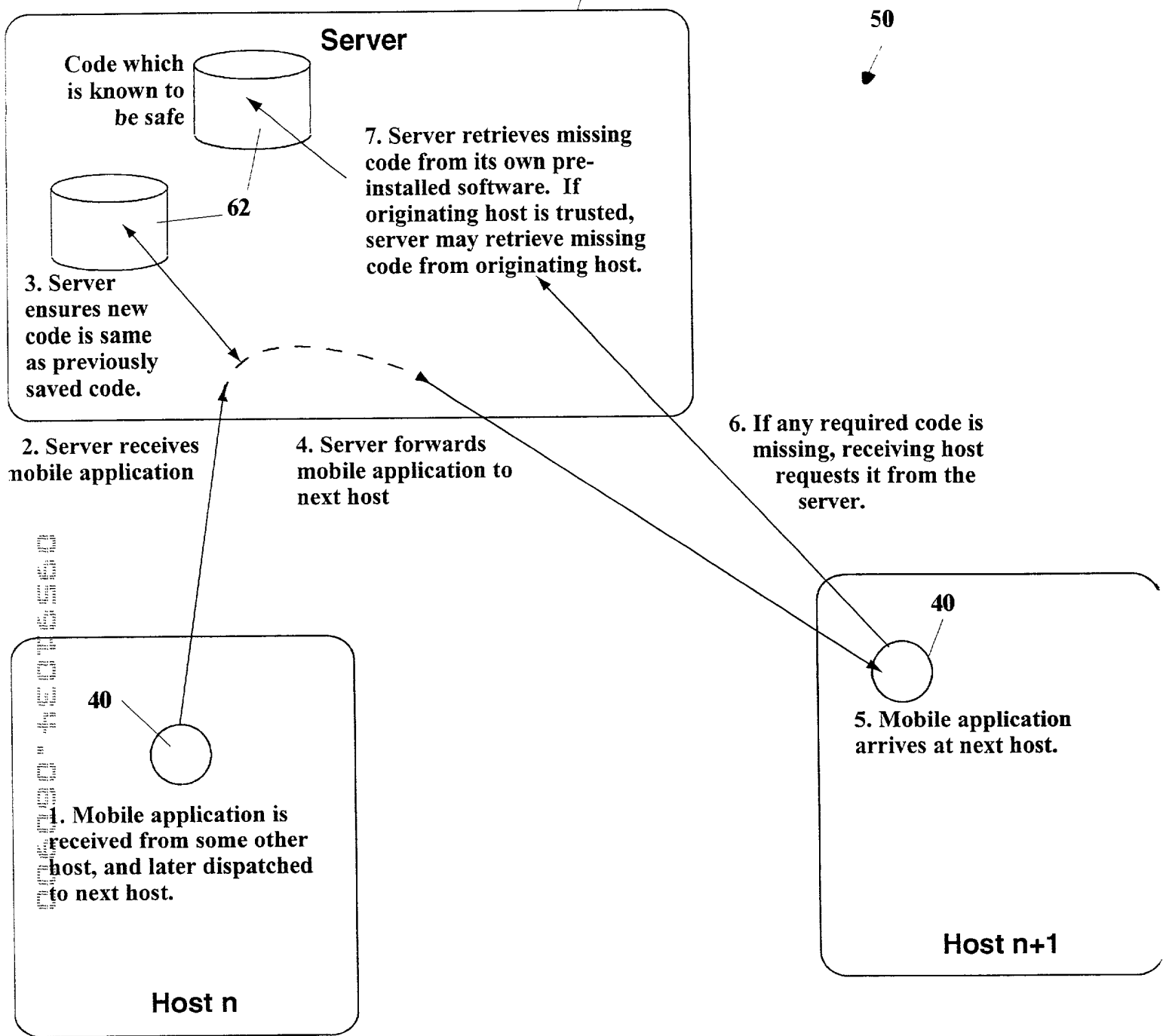


Figure 10

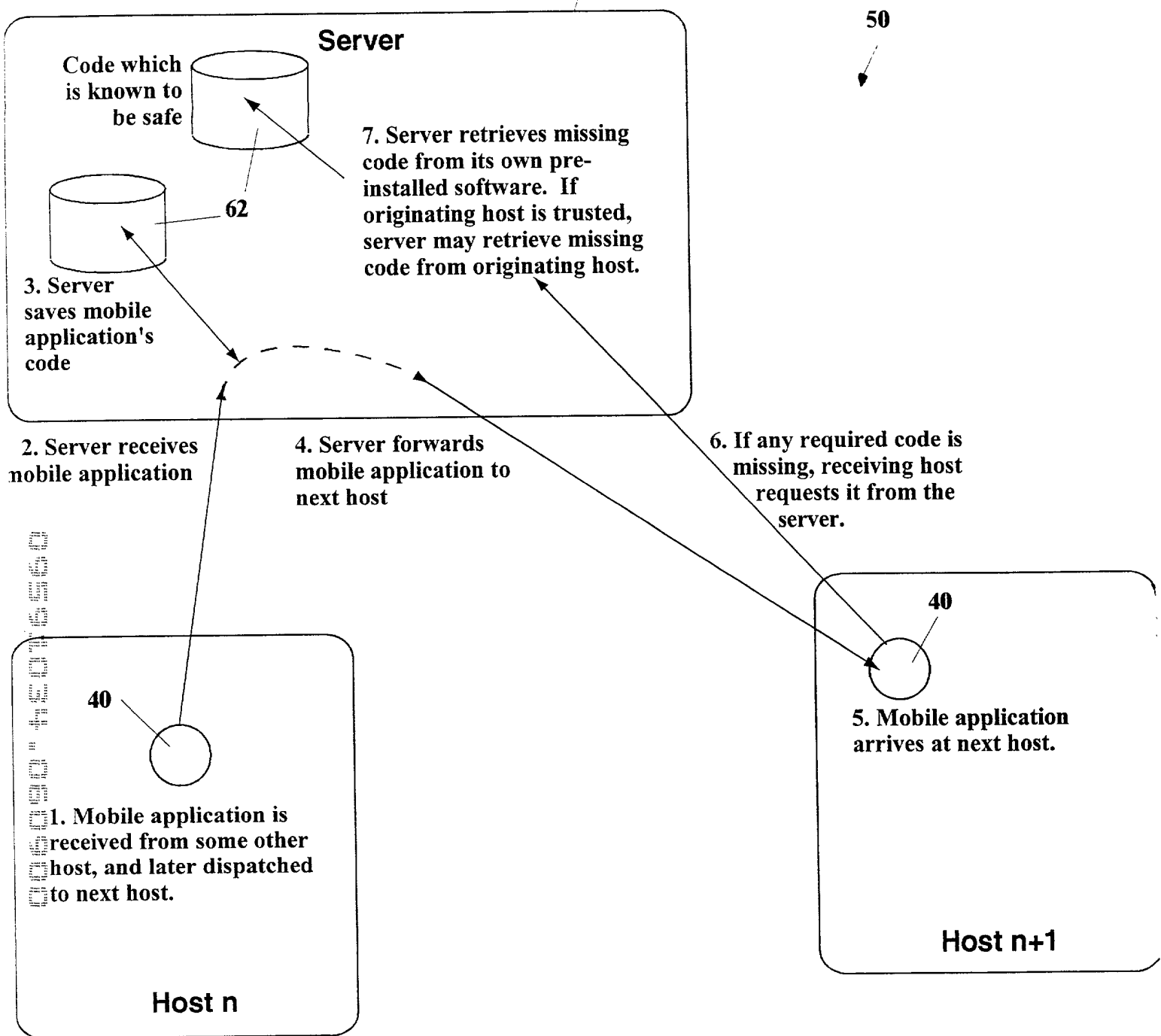


Figure 11

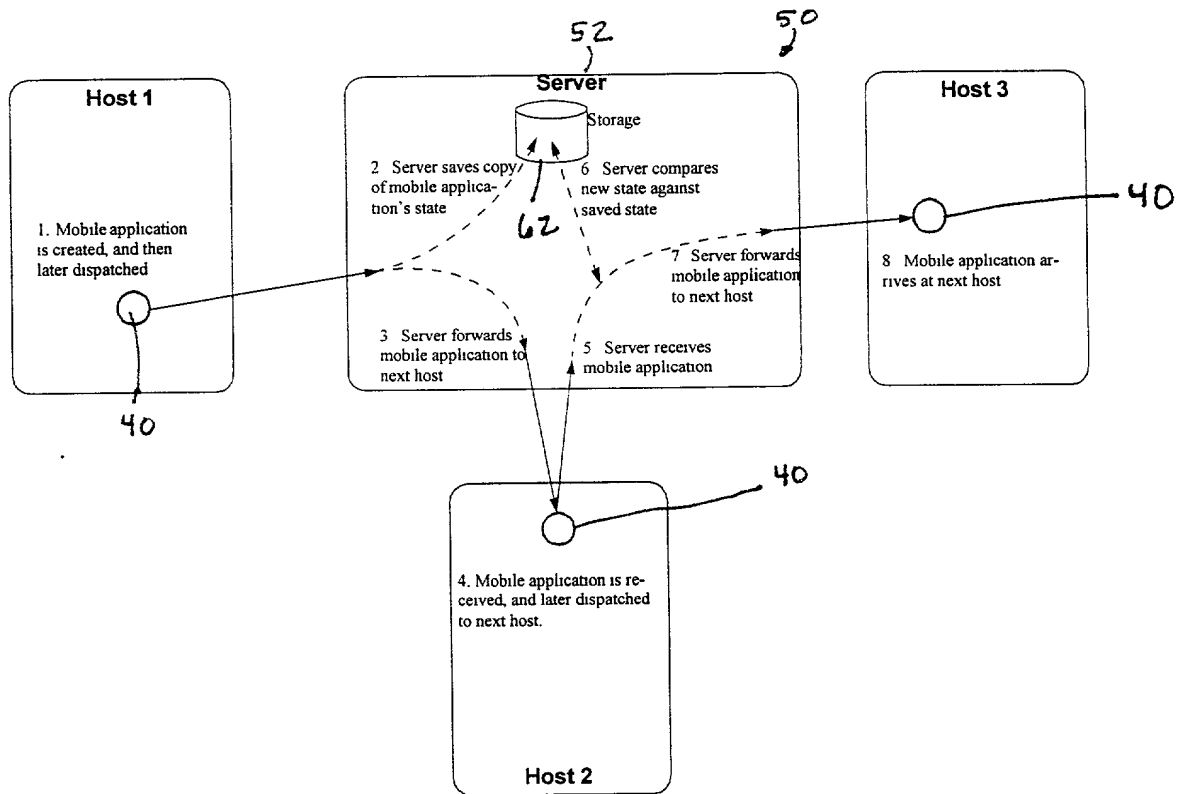


Figure 12

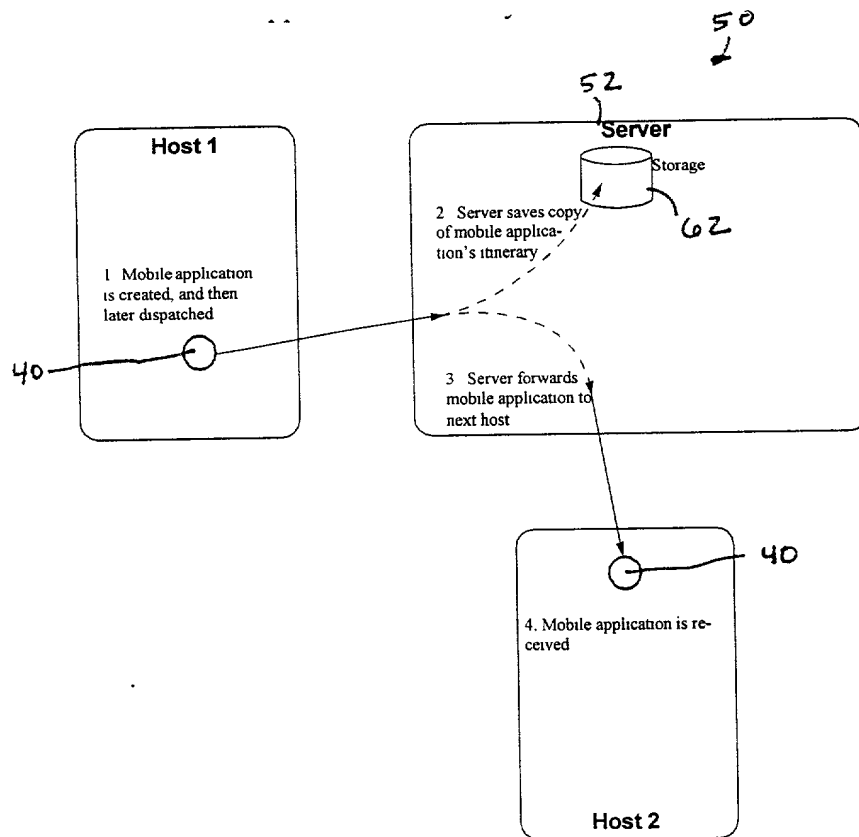


Figure 13

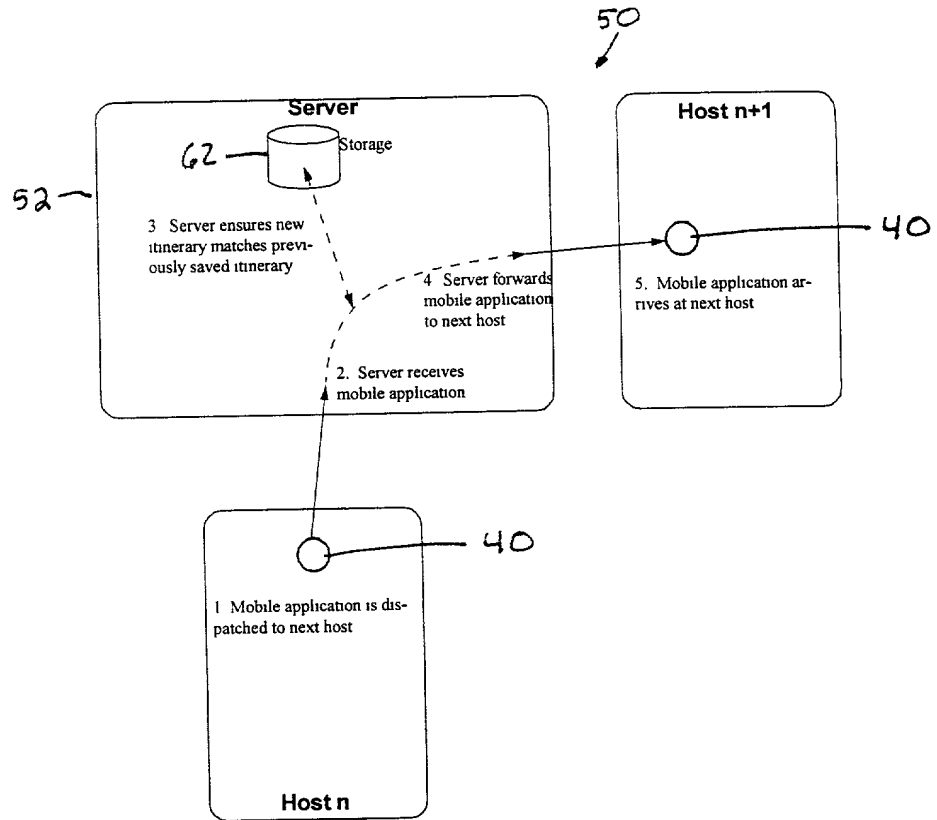


Figure 14

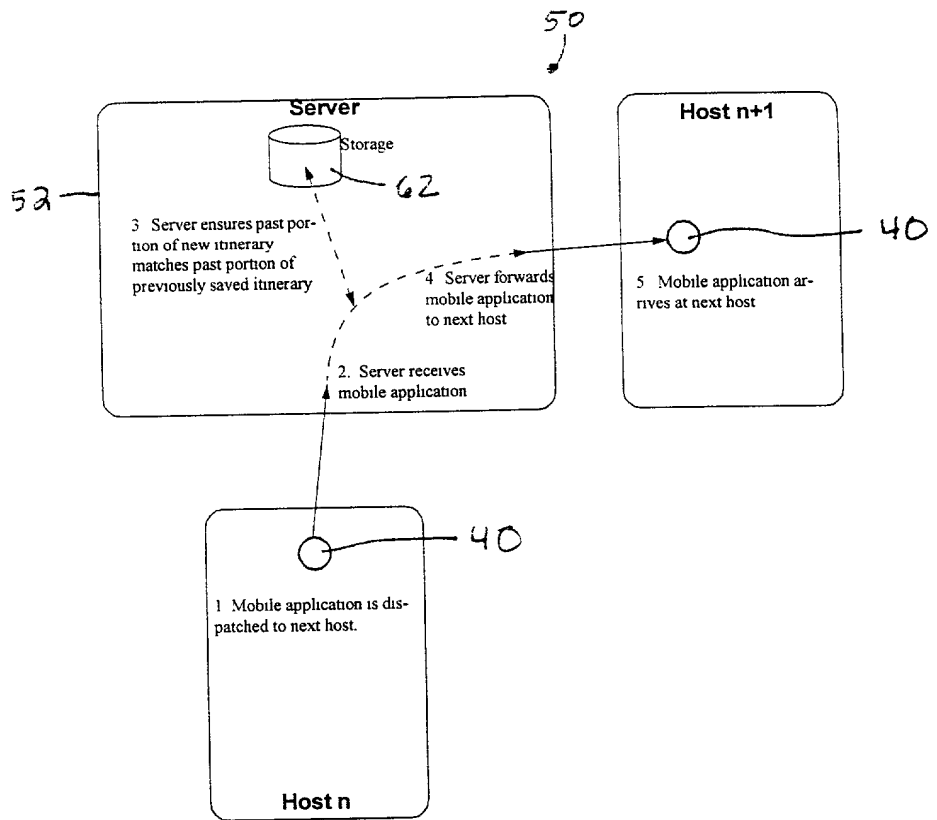


Figure 15

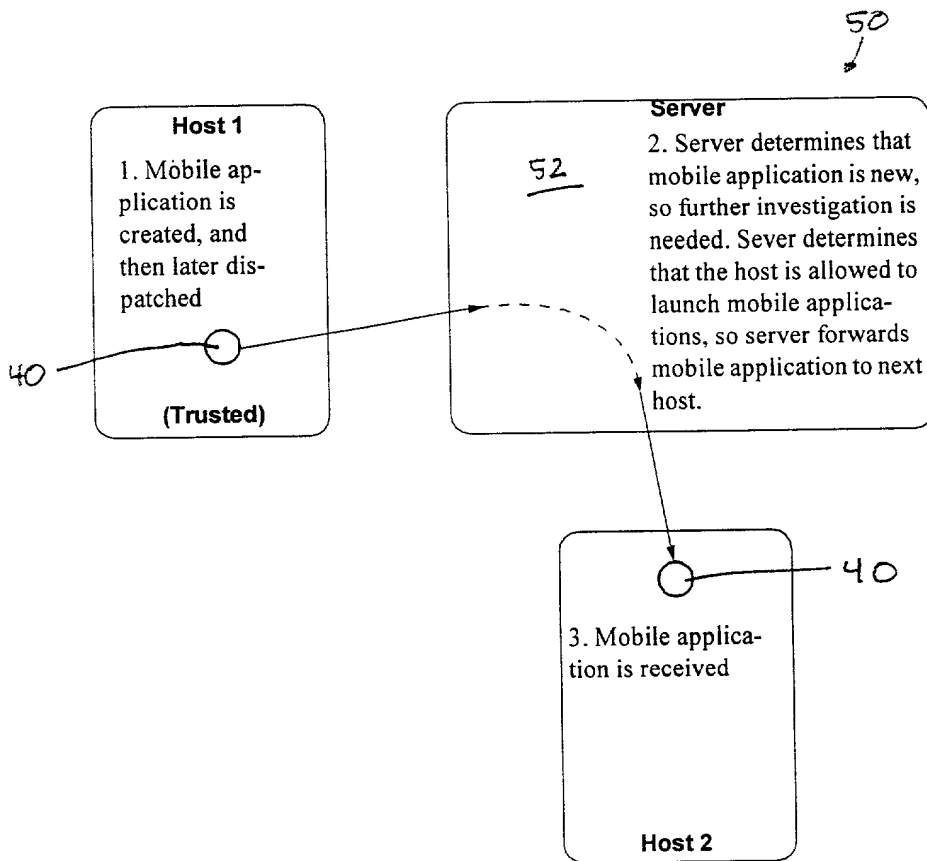


Figure 16

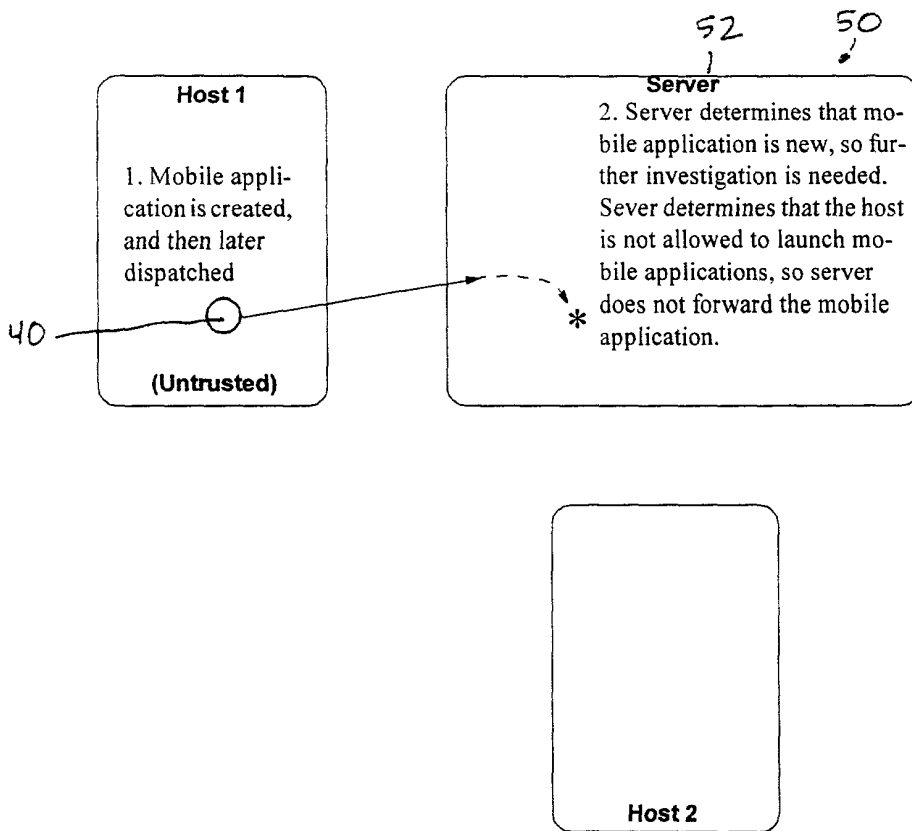


Figure 17

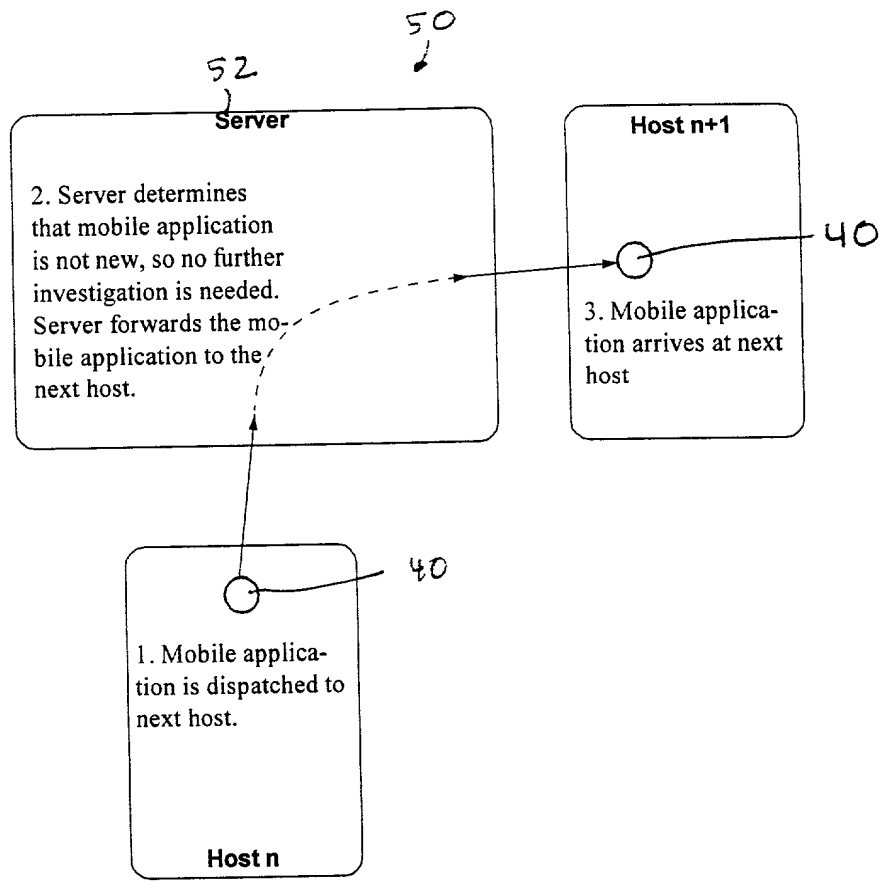


Figure 18